Copy No._____

DEFENCE R&D DÉFENSE

Implementation of a Sparse Direct Matrix Solver for the VAST Finite Element Program

L. Jiang
Martec Limited

Martec Limited
400-1888 Brunswick St.
Halifax, NS
B3J 3J8

Contract Number: W7707-010696/101/HAL

Contract Scientific Authority: M.J. Smith, 902-426-3100 x383

Defence R&D Canada

Contractor Report
DREA CR 2002-026
March 2002

National Defence Défense nationale

Canada

Copy No: _____

Implementation of a Sparse Direct Matrix Solver for the VAST Finite Element Program

L. Jiang
Martec Limited

Martec Limited
400-1888 Brunswick St.
Halifax, NS
B3J 3J8

Contract number: W7707-010696/101/HAL

Contract Scientific Authority: M.J. Smith, 902-426-3100 x383

Defence R&D Canada – Atlantic

Contract Report

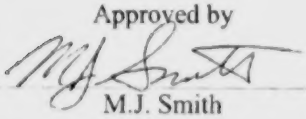
DREA CR 2002-026

March 2002

Author

L. Jiang

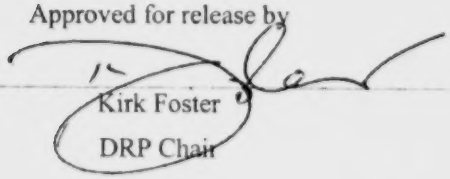
Approved by



M.J. Smith

Contract Scientific Authority

Approved for release by



Kirk Foster

DRP Chair

The scientific or technical validity of this Contract Report is entirely the responsibility of the contractor and the contents do not necessarily have the approval or endorsement of Defence R&D Canada

© Her Majesty the Queen as represented by the Minister of National Defence, 2002

© Sa majesté la reine, représentée par le ministre de la Défense nationale, 2002

Abstract

This report describes implementation of a sparse direct matrix solver, which was recently developed at DREA, into the VAST finite element program for linear and nonlinear static analyses. The procedures utilized and the new developments required in this implementation are presented in detail. Extensive numerical verifications have been conducted using test problems of various sizes. The numerical results indicated that the sparse solver produced identical finite element solutions as the skyline based direct solver, but required significantly less computation time. These benchmark problems and results are documented in this report.

Résumé

Dans ce rapport, nous présentons la mise en place d'une méthode directe de factorisation des matrices creuses linéaires, récemment conçue au CRDA, pour les analyses statiques linéaires et non linéaires par le programme d'éléments finis VAST. Nous présentons en détail les procédures utilisées et les nouveaux développements qu'a exigés l'installation de cet algorithme. Nous avons réalisé de nombreuses vérifications numériques à l'aide de problèmes tests de tailles différentes. Les résultats numériques indiquent que ce factorisateur de matrices creuses a rendu des solutions d'éléments finis identiques à celles du factorisateur direct par la méthode du profil, tout en exigeant un temps de calcul considérablement moindre. Ces tests d'évaluation de performances et leurs résultats sont décrits dans ce rapport.

104-07033/

This page intentionally left blank

Executive Summary

Background

Finite element analysis is an important component of the ongoing Improved Ship Structure Maintenance Management (ISSMM) project, which aims to provide a tool for rapid and automated risk assessment of damaged structures. A requirement of the ISSMM Software Tool (IST) now under development is that a user be able to perform a risk assessment, which may include one or more levels of finite element analysis, within a day or less. The performance of the proposed IST finite element solvers (VAST and MAESTRO) may have to be improved to meet this objective.

Principal Results

This report describes implementation of a sparse direct matrix solver that was recently developed at DREA, in the VAST finite element program for linear and nonlinear static analyses. Procedures and new developments that were required for this implementation are described in detail. Numerical verifications have been conducted using typical ship-structure test problems of various sizes. The results indicated that the sparse solver produced identical finite element solutions as the skyline based direct solver, but required significantly less computation time.

Significance of Results

The results show that the VAST sparse solver can achieve solution speeds more than ten times greater than those of the VAST skyline solver, depending on the problem. This means that both more complex models and more loading cases can now be analysed in a given time period by using the sparse solver. The benefits of this will be felt not only in the IST, but in all of DND's ship structural analysis tools that use VAST.

Future Work

The sparse solver has so far only been implemented for linear and nonlinear static finite element analysis. Similar improvement in performance can also be expected by using it for dynamic response, natural frequency and buckling analyses, and its implementation for these analysis types in VAST is recommended.

L. Jiang; 2002; Implementation of a Sparse Matrix Solver for the VAST Finite Element Program; DREA CR 2002-026; Defence Research Establishment Atlantic.

Sommaire

Cadre

L'analyse par éléments finis est une composante importante du projet de Gestion améliorée de la maintenance de la structure des navires (GAMSN) dont l'objectif est la production d'un outil d'évaluation rapide et automatique des risques que comportent des structures endommagées. Une des commandes pour l'outil logiciel GAMSN en cours d'élaboration est la possibilité d'effectuer une évaluation des risques où interviendrait un ou plusieurs niveaux d'analyse par éléments finis, en au plus une journée. Pour atteindre cet objectif, on pourrait devoir améliorer le rendement des solutionneurs d'éléments finis qui sont proposés pour l'outil logiciel GAMSN (VAST et MAESTRO).

Résultats principaux

Dans ce rapport, nous présentons la mise en place d'une méthode directe de factorisation de matrices creuses, récemment conçue au CRDA, pour les analyses statiques linéaires et non linéaires par le programme d'éléments finis VAST. Nous présentons en détail les procédures utilisées et les nouveaux développements qu'a exigés l'installation de cet algorithme. Nous avons réalisé des vérifications numériques à l'aide de problèmes tests de tailles différentes représentant une structure typique de navire. Les résultats numériques indiquent que le factorisateur de matrices creuses a donné des solutions d'éléments finis identiques à celles du factorisateur direct par la méthode du profil, tout en exigeant un temps de calcul considérablement moindre.

Importance des résultats

D'après nos résultats, le factorisateur de matrices creuses du logiciel peut résoudre certains problèmes dix fois plus rapidement que le factorisateur par la méthode de profil dans le solutionneur d'élément fini VAST. En conséquence, en utilisant le factorisateur de matrices creuses, on pourra analyser des modèles plus complexes et davantage de cas de charge dans un temps donné. Les améliorations seront ressenties non seulement dans l'utilisation de l'outil logiciel GAMSN, mais pour tous les outils d'analyse structurelle qui utilisent le VAST.

Travaux à venir

À l'heure actuelle, le factorisateur de matrices creuses n'est opérationnel que pour les analyses statiques linéaires et non linéaires par éléments finis. On peut également prévoir des améliorations similaires des résultats pour l'analyse des réactions dynamiques, des fréquences naturelles et du flambage. Nous recommandons donc l'implantation du factorisateur de matrices creuses pour ce type d'analyse dans le VAST.

L. Jiang; 2002; Implementation of a Sparse Matrix Solver for the VAST Finite Element Program (*Mise en place d'un factorisateur de matrices creuses dans le programme d'éléments finis VAST*); DREA CR 2002-026; Centre de recherches pour la défense Atlantique.

TABLE OF CONTENTS

ABSTRACT i
RÉSUMÉ i
EXECUTIVE SUMMARY iii
SOMMAIRE iv
TABLE OF CONTENTS v
LIST OF TABLES vi
LIST OF FIGURES vii
1. INTRODUCTION 1
2. IMPLEMENTATION OF SPARSE DIRECT MATRIX SOLVER IN VAST 3
3. NUMERICAL VERIFICATION AND BENCHMARK 6
3.1 Linear Static Analyses 6
3.2 Nonlinear Static Analyses 7
4. CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER WORK 18
5. REFERENCES 20

LIST OF TABLES

Table 3.1 Comparison of Computation Times Required by the Skyline and Sparse Direct Matrix Solvers for Linear Static Analyses for Test Models of Different Sizes..... 8

Table 3.2 Comparison of Computation Times Rquired by the Pre-Conditioned Conjugate Gradient Iterative Solver and the Sparse Direct Solver for Linear Analysis of PDE Engine Mount 8

Table 3.3 Comparison of Computations Time Required by the Skyline and Sparse Direct Solvers for Nonlinear Analyses of Test Models of Different Sizes..... 8

Table 4.1 Comparison of Total Computation Times Required by Skyline and Sparse Direct Matrix Solvers for Load-Shortening Curve Computation 19

LIST OF FIGURES

FIGURE 3.1 The Finite Element Mesh for Test Model *Cpfgbl* 9

FIGURE 3.2 Linear Static Deformed Configuration of Test Model *Frm43nbSag* 10

FIGURE 3.3 Linear Static Deformed Configuration of Test Model *Hopper* 11

FIGURE 3.4 Finite Element Mesh for PDE Engine Mount 12

FIGURE 3.5 Linear Static Deformed Configuration of the PDE Engine Mount 13

FIGURE 3.6 Final Deformed Configuration of a Single Stiffened Panel Resulted from a Complete Load-Shortening Curve Computation 14

FIGURE 3.7 Comparison of Load-Shortening Curves Computed Using the Skyline and Sparse Direct Matrix Solvers 15

FIGURE 3.8 Deformed Configuration of an Icebreaker Panel Under Large Displacement, Elastic-Plastic Deformation 16

FIGURE 3.9 Load-Deflection Curves at a Typical Node on the Icebreaker Panel Predicted by Skyline and Sparse Direct Matrix Solvers 17

1. INTRODUCTION

VAST is a general-purpose finite element program developed by Martec Limited over the last three decades under sponsorship of DREA [1]. The present version of VAST provides a large selection of beam, shell and solid elements based on the state-of-the-art element technology and can perform various types of engineering analyses, including static, buckling, natural frequency, shock spectrum, frequency response and dynamic time-history analyses. Nonlinearities caused by arbitrarily large displacements and rotations, large strains, elasto-plasticity and hyper-elasticity can also be considered. The VAST program has been successfully utilized to analyze numerous engineering problems and its reliability and accuracy have been well established. Recently, the VAST program has been selected as the finite element solver for the ISSMM Software Tool (IST), a DRDC Major Project now under development [2].

In addition to reliability and accuracy, numerical efficiency is also very important for a finite element program, especially when it is applied to ship structure analyses, which often involve large finite element models and numerous load cases. In recent years, several attempts have been made to improve the efficiency of the linear algebraic equation solution capability in VAST. First, in order to reduce I/O operations involved in the skyline-based matrix solver, a new module, called DECOM3, was generated, in which assembly, modification and factorization of the global stiffness matrix were combined [3]. Secondly, a commercially available matrix solver package, named DRMATHLIB, was incorporated into VAST for linear and nonlinear static analyses [4]. In this solution option, the global stiffness matrix was first assembled and modified using the VAST program and the modified global matrix was then transferred to the DRMATHLIB data format through a user interface provided by this package. Thirdly, a group of iterative solvers were implemented in VAST based on the pre-conditioned conjugate gradient algorithm [5]. Various commonly used pre-conditioners, including Jacobi, SSOR and incomplete factorization, have been provided. Numerical benchmark results indicated that these developments improved efficiency of the VAST program significantly. However, further improvements are still required.

Over the last two years, an experimental finite element program called FELAM has been developed at DREA using an object-oriented approach [6]. In this program, a direct linear algebraic equation solver was developed, based on techniques that fully exploit the sparsity of the coefficient matrix.

The node-ordering scheme used is the minimum degree algorithm, which minimizes the number of non-zero entries in the factorized coefficient matrix through a process of symbolic factorization. Early benchmark results obtained for typical ship structural finite element models indicated that the sparse direct matrix solver could be as much as an order of magnitude more efficient than the skyline direct solver in the VAST program. Recently, an out-of-core version of the sparse solver was developed, thus allowing its use for problems that exceed the physical memory of the computer.

The purpose of the present work is to incorporate DREA's out-of-core sparse direct matrix solver into the VAST program for both linear and nonlinear static analyses. Because the VAST program is written in Fortran, and the sparse direct matrix solver in C++, special interface code is required to enable data transfer between the Fortran and C++ code. Other modifications are also needed to allow the sparse direct solver to function with VAST's more extensive analysis capabilities.

In the next chapter, the procedure for implementing the sparse direct matrix solver into VAST is described, and the new code developments required in this implementation are also presented. These include the development of a new module, named DECOM5, and modifications to the existing modules, DISP1 and DISPFV. As mentioned above, a set of special-purpose interface routines were also developed to ensure proper data communication between the Fortran and C++ code. These interface routines are included in a new module, named VASTSP.

In order to verify the accuracy and efficiency of the newly implemented sparse direct matrix solver, a large number of linear and nonlinear finite element problems of different sizes have been considered. These test problems were solved using the new sparse matrix solver and the various older matrix solvers providing in the VAST program, such as the skyline-based direct solver and the iterative solver based on the pre-conditioned conjugate gradient algorithm. A comparison of the numerical results and computation times indicated that the sparse solver produced identical solutions as the existing solvers, but required significantly less execution time. The benchmark problems and results are documented in Chapter 3 in detail.

The concluding remarks and recommendations for further work are given in Chapter 4.

2. IMPLEMENTATION OF SPARSE DIRECT MATRIX SOLVER IN VAST

To ensure efficient implementation of the sparse direct solver into VAST, a new module, named DECOM5, was first developed. This development was based on an existing module, DECOM3, which was developed earlier to enhance the performance of the skyline direct matrix solver. In DECOM5, statistical information of a finite element problem, such as total number of displacement nodes (NDN), number of degrees of freedom per node (NDF), and total number of elements (NELM) *et al.*, were first retrieved from the binary intermediate files generated during element stiffness calculations. Boundary conditions, including boundary springs, multi-point constraint (MPC) equations and rigid links, were then read from the input data file. If the penalty method is utilized for treating the multi-point constraints and rigid links, penalty stiffness matrices must be computed. After all these information become available, subroutines from the sparse direct matrix solver package are called to perform assembly, modification and factorization of the global stiffness matrix. In the sparse solver option, element stiffness matrices are stored in a random-access file, which ensures the efficiency of assembly operations.

Because the assembly algorithm adopted in the sparse direct matrix solver requires that the element stiffness matrices be formulated in terms of nodes rather than degrees-of-freedom, some special considerations are required for the penalty stiffness matrix formulation. Expressing the MPC equations in matrix form, we have:

$$\begin{bmatrix} \boldsymbol{G}_i & \boldsymbol{G}_j \end{bmatrix} \begin{Bmatrix} \boldsymbol{U}_i \\ \boldsymbol{U}_j \end{Bmatrix} = \boldsymbol{0} \tag{1}$$

where $\boldsymbol{U}_i$ and $\boldsymbol{U}_j$ indicate the master and slave degrees-of-freedom, respectively. $\boldsymbol{G}_i$ and $\boldsymbol{G}_j$ are matrices which contain coefficients in the multi-point constraint equations. For rigid links, the $\boldsymbol{G}_i$ and $\boldsymbol{G}_j$ matrices are updated continuously during nonlinear analyses using the most updated problem geometry. With the above MPC equations, the penalty stiffness matrix can be readily formulated as:

$$\boldsymbol{K}_p = \begin{bmatrix} \boldsymbol{G}_i^T \\ \boldsymbol{G}_j^T \end{bmatrix} A \begin{bmatrix} \boldsymbol{G}_i & \boldsymbol{G}_j \end{bmatrix} \tag{2}$$

where A is a diagonal matrix containing the penalty parameters. If the sparse direct solver is selected, this penalty stiffness matrix must be expended to include complete set of nodal degrees-of-freedom for all the nodes involved in the MPC equations.

In addition to factorization, subroutines from the sparse matrix solver were also needed to be implemented into the DISP1 module to perform forward reduction and backward substitution. For linear analysis, the implementation was straightforward, because it just required solution for a single right hand side vector R as:

$$U = K^{-1}R \tag{3}$$

where U denotes the displacement vector and K the global stiffness matrix. For nonlinear analysis, however, the implementation of the sparse solver was slightly more complicated because the use of the advanced solution algorithm for nonlinear finite element problems, such the arc-length method, required two solutions of the linearized system in each equilibrium iteration. For instance, writing the linearized equilibrium equations as:

$$^{t+\Delta t}K^{(k-1)}\Delta U^{(k)} = \left(^{t+\Delta t}\lambda^{(k-1)} + \Delta\lambda^{(k)}\right)R - {}^{t+\Delta t}f_{\text{int}}^{(k)}\,, \tag{4}$$

intermediate solution vectors were first computed as:

$$\begin{aligned} Y &= {}^{t+\Delta t}K^{(k-1)^{-1}}R \\ Z &= {}^{t+\Delta t}K^{(k-1)^{-1}}\left({}^{t+\Delta t}\lambda^{(k-1)}R - {}^{t+\Delta t}f_{\text{int}}^{(k)}\right) \end{aligned} \tag{5}$$

and the solution vector was expressed as a function of the incremental load parameter as:

$$\Delta U^{(k)} = Y\Delta\lambda^{(k)} + Z\,. \tag{6}$$

This relationship was subsequently used in the solution step control equation, such as the constant arc-length equation, to determine the incremental displacements and incremental load parameter, simultaneously. In the above equations, $t+\Delta t$ and (k) indicate the current time step and iteration number, respectively. f_{int} and R denote the internal force vector and reference external load vector.

In addition to the regular displacement solution module DISP1, the sparse matrix solver has also been implemented into a fast version of displacement module, DISPFV, which are specially developed for handling large numbers of load cases. In this module, multiple load cases are solved simultaneously, and the number of load cases can be considered in each solution pass depends on the memory available. Separate subroutines from the sparse direct solver are utilized for treating the situation of multiple load vectors.

In the present implementation, the sparse direct matrix solver can be simply activated by setting the main solution control parameter ISOLVE=4 and the master control code IDECOM=5. In addition, master control codes for assembly, IASSEM, and stiffness matrix modification, ISTIFM, must be set to zero.

3. NUMERICAL VERIFICATION AND BENCHMARK

In order to verify the correctness and benchmark the efficiency of the newly implemented sparse direct matrix solver in VAST, an extensive numerical study has been conducted using a large group of test example problems involving linear and nonlinear static analyses. For all the test problems, numerical results produced by the new sparse solver and the various existing solution algorithms were compared and exact agreement between these solutions was identified. In the following section, computation times required by the various linear algebraic equation solvers are compared for the test problems.

3.1 Linear Static Analyses

Six test problems involving linear static analyses of shell structures were first considered. These models were initially generated at DREA for testing the C++ finite element program FELAM [6] and were provided to Martec for the present contract. The sizes of these models vary widely from 500 and 25000 nodes, as summarized in Table 3.1. For the full CPF model *Cpfgbl* shown in Figure 3.1, as many as 207 load cases were considered. For all the other test models, linear elastic displacements and stresses were computed for single load vector. The computation times required by the skyline-based and sparse direct matrix solvers for factorization (DECOM) and forward reduction/backward substitution (DISP) are compared in Table 3.1. The percentage savings gained by using the sparse solver are shown in the last column of the table. These results clearly demonstrate the efficiency of the sparse direct solver. They also indicated that the benefit of using the sparse solver is problem-dependent. For huge finite element models of complicated shell structures, for which the bandwidth reduction algorithm is unable to effectively reduce the total number of entries below the skyline, the sparse direct matrix solver can be a magnitude more efficient than the skyline-based solver. The predicted deformed configurations of the two largest test models, *Frm43nbSag* and *Hopper*, are displayed in Figures 3.2 and 3.3.

As has been reported before, although the pre-conditioned conjugate gradient solver implemented in VAST did not perform satisfactorily for shell-like structures, it was very efficient for problems having comparable dimensions in all three directions, such as a solid cube modeled by 3D brick elements [5]. To compare efficiency of the sparse direct solver and the iterative solver of this type of problems, an 8-noded solid element model of the PDE engine mount was considered. The

engine mount contains steel top and bottom plates and a soft rubber core, which are shown in red and blue, respectively in Figure 3.4. Linear static solution of the mount under lateral load was obtained by using both the sparse direct matrix solver and the pre-conditioned conjugate gradient solver. The deformed configuration predicted by both solvers was identical and is shown in Figure 3.5. In the solution using the iterative solver, pre-conditioner based on the symmetric successive over-relaxation method (SSOR) was employed. The solution times required by the sparse direct solver and the pre-conditioned conjugate gradient solver are compared in Table 3.2. For this particular problem, the iterative solver was more efficient. It should be mentioned at this point that the skyline-based solver failed for this problem due to the huge memory requirement of the bandwidth reduction computation.

3.2 Nonlinear Static Analyses

To benchmark the sparse direct solver for nonlinear analyses, two test example problems were considered. The first problem involved computation of complete load-shortening curve of a typical single stiffened panel, and the second problem involved large displacement, elastic-plastic analysis of an icebreaker girder subjected to lateral pressure load. The sizes of these shell element models are given in Table 3.3.

For both nonlinear problems, identical results were obtained by using the skyline-based and sparse direct solvers. The final deformed shape of the single stiffened panel is displayed in Figure 3.6. The load-shortening curves predicted by the skyline and sparse solvers were identical as depicted in Figure 3.7. For the icebreaker girder, the predicted final deformed configuration and load-deflection curves at a typical node are shown in Figures 3.8 and 3.9, respectively. The computation times for solving linear algebraic equations in both solver options are compared in Table 3.3. The times reported in this table are the total times required for the entire nonlinear analysis. For the single stiffened panel problem, there were 48 solution steps and a total of 149 iterations. For the icebreaker problem, nonlinear solutions were obtained for the first 10 steps and a total of 30 iterations were taken to achieve convergence. For both problems, the sparse matrix solver is considerably more efficient than the skyline-based solver and the percentage savings are consistent with those observed in linear analyses. In addition, the successful computation of the complete load-shortening curve using the sparse matrix solver confirmed its ability to handle linear algebraic

equation systems having non-positive definite coefficient matrices.

Table 3.1 Comparison of Computation Times Required by the Skyline and Sparse Direct Matrix Solvers for Linear Static Analyses for Test Models of Different Sizes.

Model Name	# of Nodes	Skyline Solver (sec.)		Sparse Solver (sec.)		Savings (%)
		DECOM3	DISP	DECOM5	DISP	
LSPCurve	527	0.66	0.00	0.61	0.05	0.00
Detail1	4929	113.97	6.70	17.30	0.39	85.34
Frame31Hog	7160	115.9	5.49	33.23	0.44	72.26
Cpfgbl	7534	116.11	417.38	127.21	65.25	63.92
Frm43nbSag	10645	442.70	57.89	66.13	0.77	86.64
Hopper	25154	7237.36	487.80	449.07	25.43	93.86

Table 3.2 Comparison of Computations Time Required by the Pre-Conditioned Conjugate Gradient Iterative Solver and the Sparse Direct Solver for Linear Analysis of PDE Engine Mount.

Model Name	# of Nodes	PCG Solver (sec.)		Sparse Solver (sec.)	
		DECOM7	DISP	DECOM5	DISP
PDEmount	14424	10.05	261.50	1068.85	42.67

Table 3.3 Comparison of Computations Time Required by the Skyline and Sparse Direct Solvers for Nonlinear Analyses of Test Models of Different Sizes.

Model Name	# of Nodes	Skyline Solver (sec.)		Sparse Solver (sec.)		Savings (%)
		DECOM3	DISP	DECOM5	DISP	
LSpanel	1211	585.59	41.96	413.11	18.77	31.18
IceBreaker	7343	6864.64	699.88	1366.45	24.99	81.61

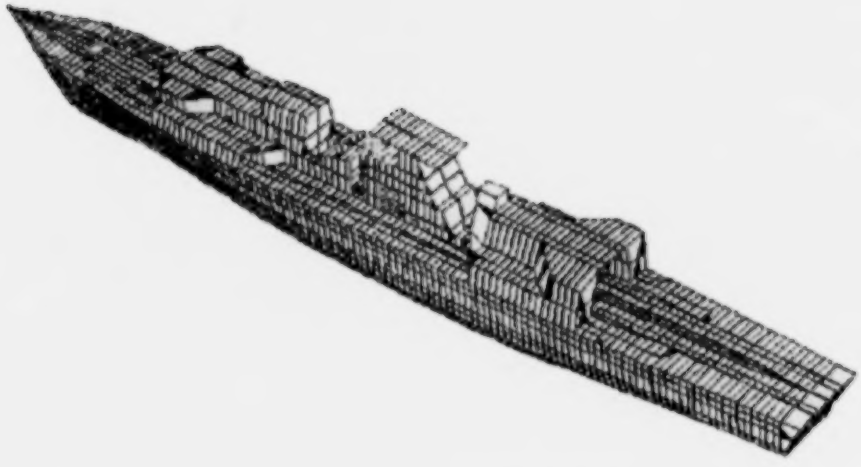


FIGURE 3.1: The Finite Element Mesh for Test Model *Cpfgbl*

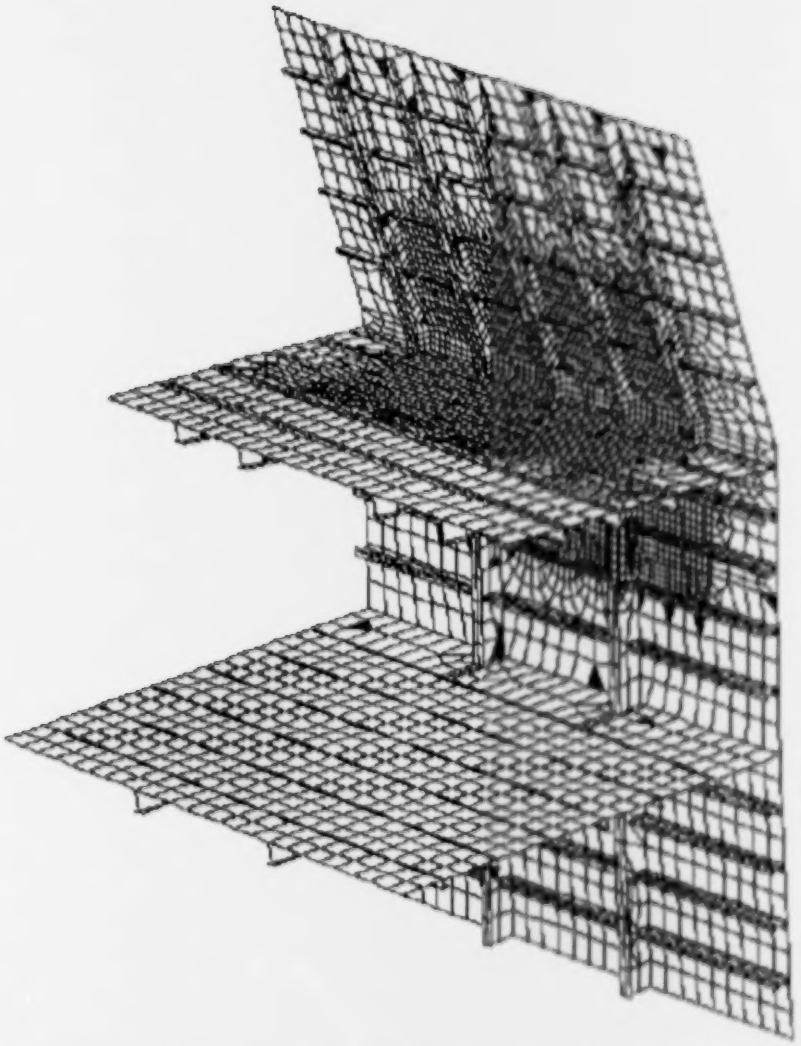


FIGURE 3.2: Linear Static Deformed Configuration of Test Model *Frm43nbSag*

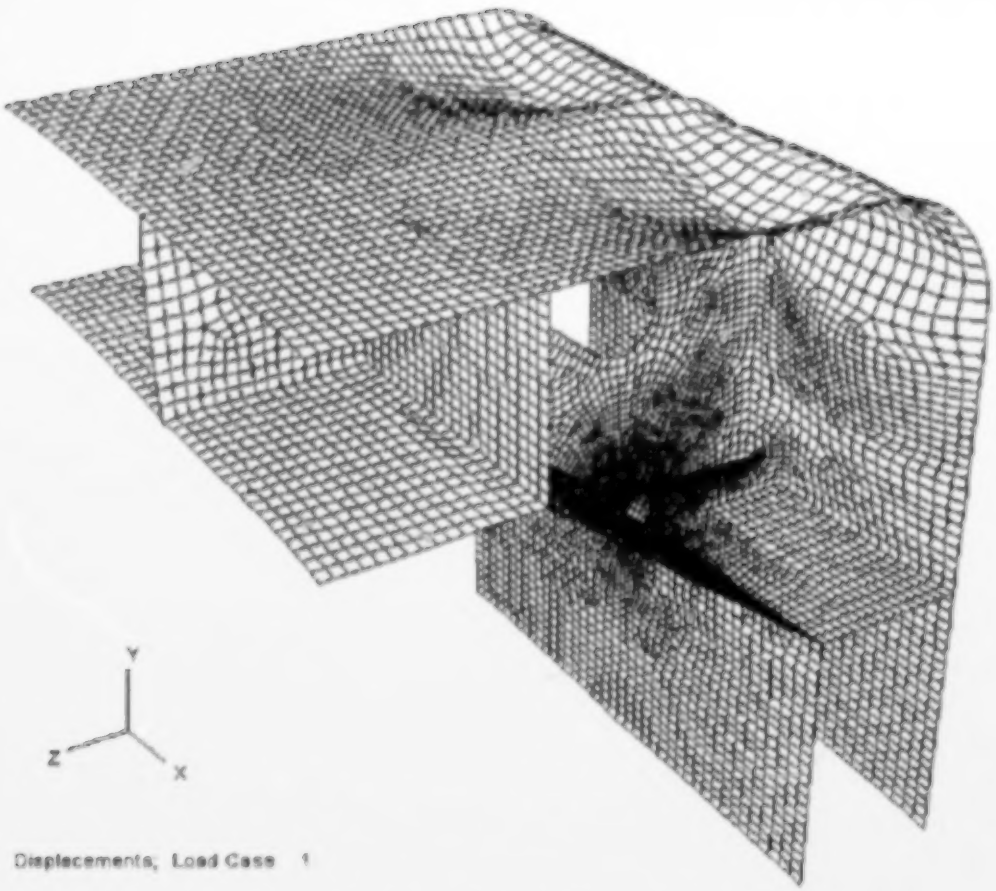


FIGURE 3.3: Linear Static Deformed Configuration of Test Model *Hopper*.

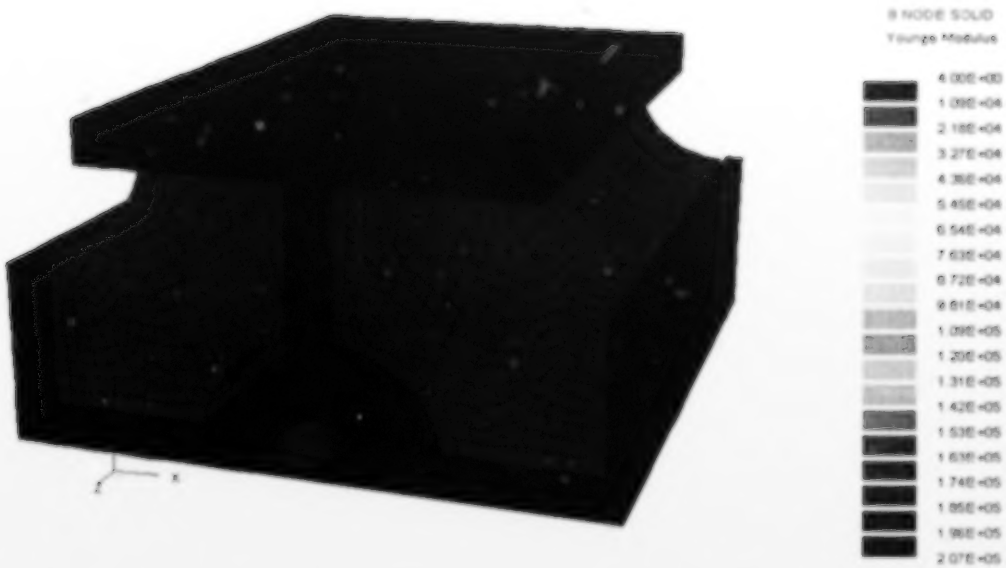


FIGURE 3.4: Finite Element Mesh for PDE Engine Mount

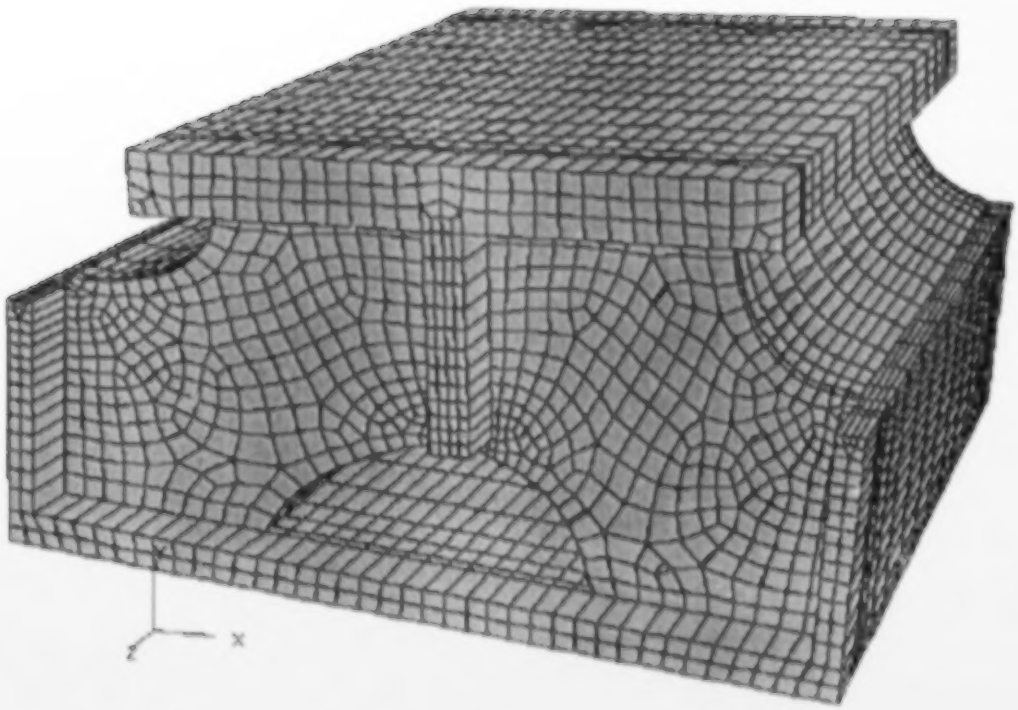


FIGURE 3.5: Linear Static Deformed Configuration of the PDE Engine Mount.

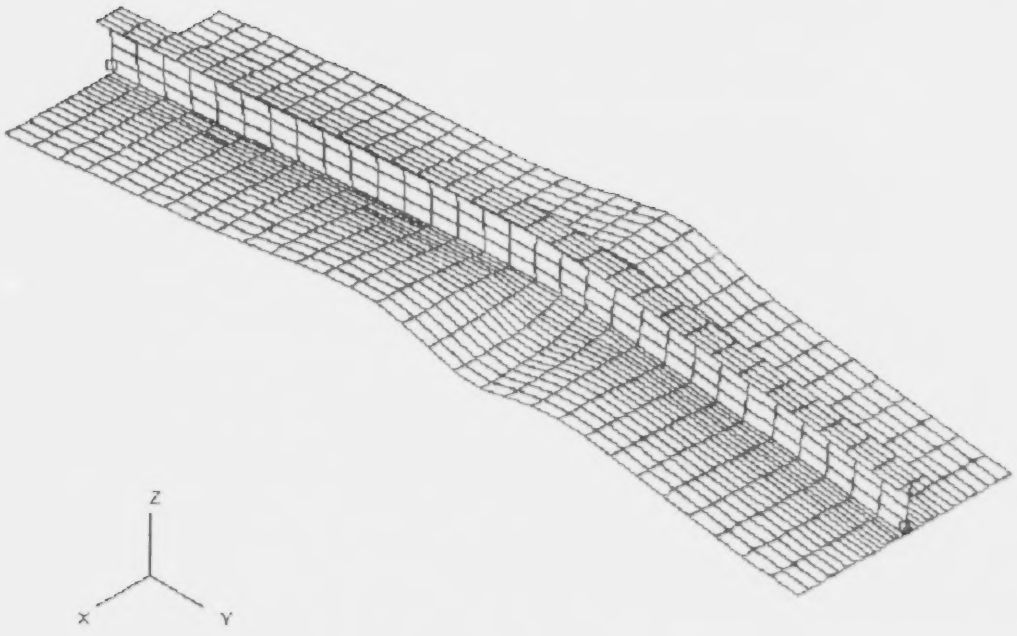


FIGURE 3.6: Final Deformed Configuration of a Single Stiffened Panel Resulted from a Complete Load-Shortening Curve Computation

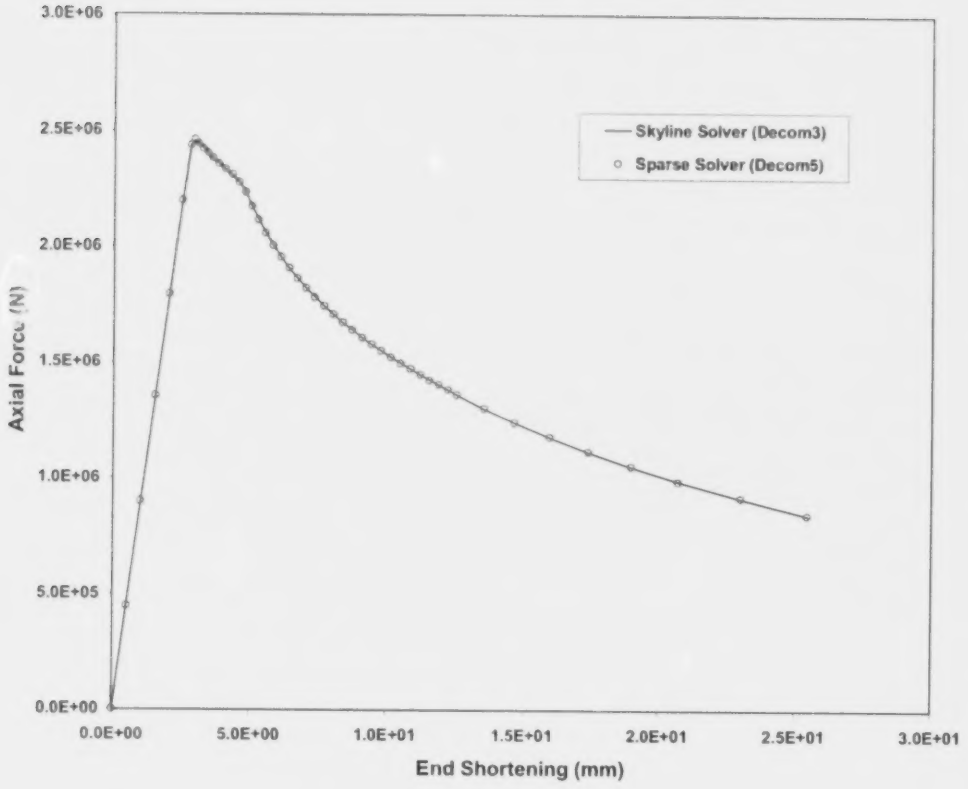


FIGURE 3.7: Comparison of Load-Shortening Curves Computed Using the Skyline and Sparse Direct Matrix Solvers

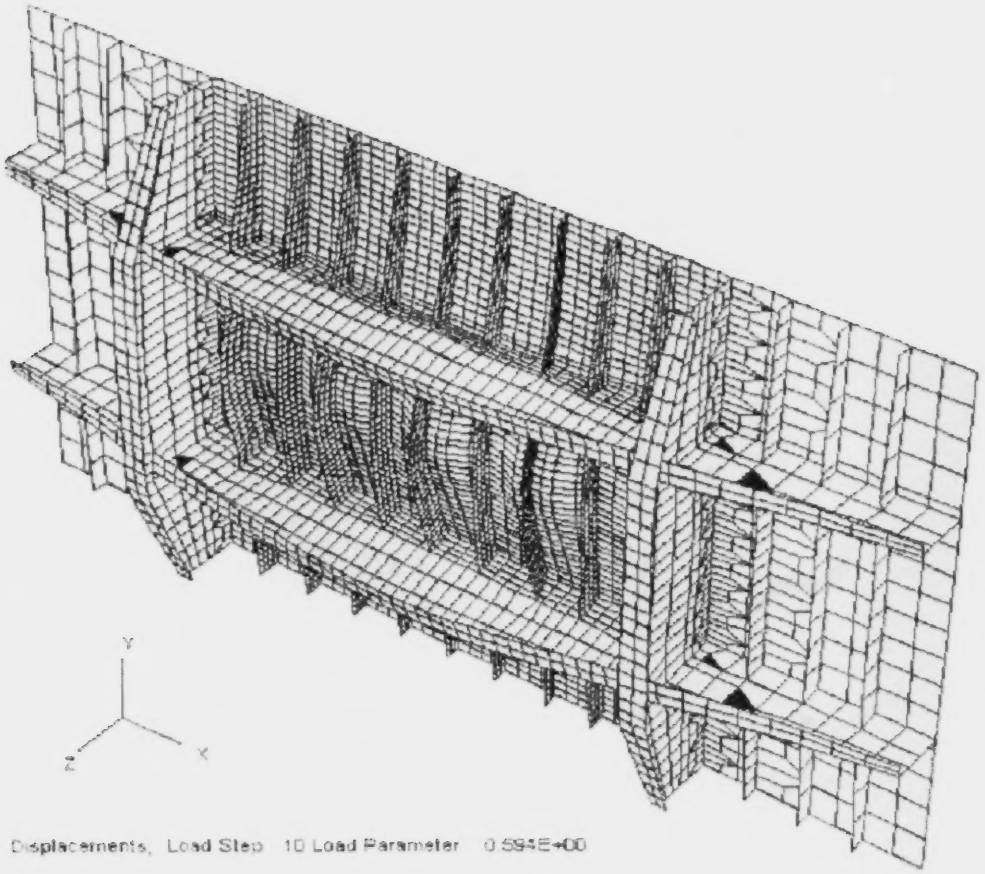


FIGURE 3.8: Deformed Configuration of an Icebreaker Panel under Large Displacement, Elastic-Plastic Deformation

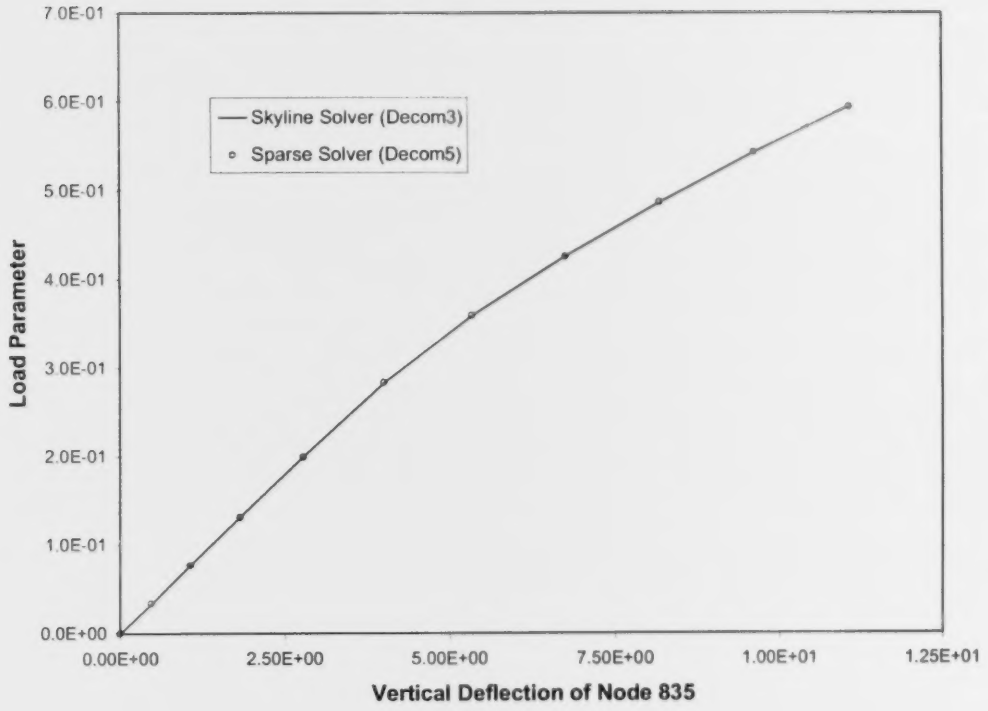


FIGURE 3.9: Load-Deflection Curves at a Typical Node on the Icebreaker Panel Predicted by Skyline and Sparse Direct Matrix Solvers

4. CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER WORK

In this report, we have described a recent implementation and verification of a sparse direct matrix solver in the VAST finite element program. The sparse direct solver was developed at DREA and has been incorporated in VAST as an external linear algebraic equation solver package to perform linear and nonlinear static analyses. In order to ensure seamless connection between VAST and the sparse matrix solver, a new module, named DECOM5, was created and the existing modules DISP1 and DISPFV were modified. To verify the correctness and benchmark the efficiency of the newly implemented sparse matrix solver, extensive numerical studies have been carried out using a group of carefully selected linear and nonlinear problems. A comparison of the numerical results and computation times for using various solver options provided by VAST confirmed that the sparse direct matrix solver did produce correct solutions, even in the post-buckling range where the tangent stiffness matrix already became non-positive definite. The sparse matrix solver can be as much as one magnitude more efficient than the skyline-based solver, depending on the size and complexity of the problem.

The numerical results obtained in this work indicated that the sparse direct matrix solver is very promising. However, the present implementation is only limited to linear and nonlinear static analyses and it should definitely be extended to eigenvalue and dynamic problems. The finite element procedures for eigenvalue and dynamic problems are more complicated. In addition to solving systems of linear algebraic equations, repeated matrix-vector multiplication must also be performed. As a result, in order to ensure numerical efficiency of the sparse solver in eigenvalue and dynamic analyses, additional capabilities must be developed in the sparse solver package to carry out this multiplication in a highly efficient manner. To this end, the assembled global stiffness and mass matrices may need to be stored using the sparse matrix format.

Due to the speed up of the VAST modules involved in solutions of linear algebraic equations, the modulus for element and stress calculations took a much more significant portion of the total computation time, especially for nonlinear analyses of finite element models of moderate size. An example is given in Table 4.1. As a result, in order to further reduce the total computation time, we have to also improve the efficiency of element formulation and stress calculation in VAST, in addition to further speed up of the linear algebraic equation solver.

Table 4.1 Comparison of Total Computation Times Required by Skyline and Sparse Direct Matrix Solvers for Load-Shortening Curve Computation.

VAST Module	Skyline Solver		Sparse Solver	
	Time (Sec.)	Percentage (%)	Time (Sec.)	Percentage (%)
Element Formulation	314.07	23.81	314.61	33.89
Matrix Factorization	825.11	62.54	462.07	49.77
Displacement Computation	59.33	4.50	25.89	2.78
Stress Calculation	120.76	9.15	125.89	13.56
Total	1319.27	100.00	928.46	100.00

5. REFERENCES

[1] MARTEC LIMITED. "Vibration and Strength Analysis Program (VAST): User's Manual Version 8.2". Martec Limited, Halifax, 2000.

[2] PEGG, N. G. "Development of the ISSMM Software Tool for Residual Strength Assessment of Ship Structures". (DREA TM 1999-055). Defence Research Establishment Atlantic, 1999.

[3] JIANG, L., NORWOOD, M. E., CHERNUKA, M. W., and COVILL, J. D. "Pilot Study in Improving Efficiency of the VAST Finite Element Program". Martec Technical Report TR-96-19, Martec Limited, Halifax, 1996.

[4] JIANG, L., and CHERNUKA, M. W. "IVAST-An Efficient VAST Finite Element Program for Linear and Nonlinear Static Analysis". Martec Technical Report TR-98-05, Martec Limited, Halifax, 1998.

[5] JIANG, L., and CHERNUKA, M. W. "Improved Nonlinear Solution Techniques in the VAST Finite Element Program". Martec Technical Report TR-99-32, Martec Limited, Halifax, 2000.

[6] SMITH, M. J. "First-Ply Failure Prediction of Fibrous Composites with Finite Element Analysis". (DREA TM 2001-065). Defence Research Establishment Atlantic, 2001.

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Establishment sponsoring a contractor's report, or tasking agency, are entered in section 8.)

Defence Research Establishment Atlantic
PO Box 1012
Dartmouth, NS, Canada B2Y 3Z7

2. SECURITY CLASSIFICATION (overall security classification of the document including special warning terms if applicable)

UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C,R or U) in parentheses after the title).

Implementation of a Sparse Matrix Solver for the VAST Finite Element Program

4. AUTHORS (Last name, first name, middle initial. If military, show rank, e.g. Doe, Maj. John E.)

Jiang, L.

5. DATE OF PUBLICATION (month and year of publication of document)

March 2002

6a. NO. OF PAGES (total containing information Include Annexes, Appendices, etc).

27 (approx.)

6b. NO. OF REFS (total cited in document)

6

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered).

Contractor Report

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include address).

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant).

9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written).

W7707-010696/001/HAL

10a ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document.)

DREA CR-2002-026

10b OTHER DOCUMENT NOs. (Any other numbers which may be assigned this document either by the originator or by the sponsor.)

Martec Technical Report TR-02-07

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

(X) Unlimited distribution
( ) Defence departments and defence contractors; further distribution only as approved
( ) Defence departments and Canadian defence contractors; further distribution only as approved
( ) Government departments and agencies; further distribution only as approved
( ) Defence departments; further distribution only as approved
( ) Other (please specify): Limited to MARIN CRS member agencies only

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in (11) is possible, a wider announcement audience may be selected).

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual)

This report describes implementation of a sparse direct matrix solver, which was recently developed at DREA, into the VAST finite element program for linear and nonlinear static analyses. The procedures utilized and the new developments required in this implementation are presented in detail. Extensive numerical verifications have been conducted using test problems of various sizes. The numerical results indicated that the sparse solver produced identical finite element solutions as the skyline based direct solver, but required significantly less computation time. These benchmark problems and results are documented in this report.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title).

Ship structure
Finite element analysis
Sparse solver
Matrix factorisation
VAST